WEST

Generate Collection Print

L4: Entry 3 of 28 File: USPT

US-PAT-NO: 6243735
DOCUMENT-IDENTIFIER: US 6243735 B1

TITLE: Microcontroller, data processing system and task switching control method

DATE-ISSUED: June 5, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Imanishi; Hiroshi	Kyoto			JP
Araki; Toshiyuki	Osaka			JP

US-CL-CURRENT: 709/102; 709/100

CLAIMS:

What is claimed is:

1. A microcontroller comprising:

(a) a processor for sequentially executing a plurality of tasks in accordance with programmed instructions, said processor operating in conjunction with a plurality of hardware engines;

(b) a task management table for storing task management information including (1) state information representative of the execution status of each said task, (ii) priority information representative of the execution priority of each said task, and (iii) allocation information representative of the allocation of said plurality of tasks to said plurality of hardware engines; and

(c) a scheduler for allowing, on the basis of said task management information, said processor to switch between tasks, wherein each said plurality of hardware engines starts execution of a data process upon the activation by said processor and, if said data process is terminated, informs said scheduler of the termination of execution, and said scheduler allows said processor to switch between tasks if the termination of execution of any one of said hardware engines is detected.

2. The microcontroller according to claim 1, wherein each said task can be in one of a first state representative of an execution wait status, a second state representative of a running status, and a third state representative of a wait status awaiting the termination of execution of a hardware engine allocated thereto.

3. The microcontroller according to claim 2, wherein when during execution of a task of said plurality of tasks said processor activates a hardware engine allocated to said task under execution before decoding a given instruction, said processor performs a function of updating said state information so that said task makes a state transition from said second state to said third state.

4. The microcontroller according to claim 2,

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3286239	November 1966	Thompson et al.	340/172.5
☐	3312951	April 1967	Hertz	340/172.5
☐	3333252	July 1967	Shimabukuro	340/172.5
☐	3359544	December 1967	Macon et al.	340/172.5
☐	3363234	January 1968	Erickson et al.	340/172.5
☐	3399384	August 1968	Crockett et al.	340/172.5
☐	3421150	January 1969	Quosig et al.	340/172.5
☐	3449722	June 1969	Tucker	340/172.5
☐	3491339	January 1970	Schramel	340/172.5
☐	3496551	February 1970	Driscoll et al.	340/172.5
☐	3530438	September 1970	Mellen et al.	340/172.5

ART-UNIT: 237

PRIMARY-EXAMINER: Shaw; Gareth D.

ASSISTANT-EXAMINER: Chirlin; Sydney R.

ABSTRACT:

A program scheduler is provided for use with a multiprocessor system or its equivalent, such as a multiprogrammed processor unit, and the program scheduler receives tasks to be executed, schedules them for assignment, allots a task to each processor and interrupts the processors to assign new tasks. The program scheduler includes a plurality of buckets or tables where task words are stored, and associated with each task word is a T.sub.e field which specifies the estimated processor time required to complete the task and a T.sub.d field which indicates the time remaining before the task must be completed. The ratio T.sub.e /T.sub. d provides an indication of the need of each task word for processor service since the need for such service becomes more urgent as the ratio approaches 1. A scheduling algorithm periodically recalculates the service ratio and shifts tasks, if need be, from one table to another whereby tasks with a similar service ratio are stored in a common table. Task words within a given table are divided into classes according to the length of time a task has not received service. An allocation algorithm allots tasks to processors from the older classes first and proceeds in sequence through the various classes to the latest classes. Both the scheduling algorithm and the allocation algorithm service all tables in the program scheduler, but the tables with higher service ratios are serviced more often by each algorithm than tables with lower service ratios. When many task words are awaiting processor service, a given task word receives processor service at a rather low frequency when it has a small service ratio, but it receives processor service at a relatively high frequency as its service ratio approaches 1.

40 Claims, 65 Drawing figures

WEST

End of Result Set

Generate Collection Print

L4: Entry 28 of 28 File: USPT Mar 7, 1972

US-PAT-NO: 3648253
DOCUMENT-IDENTIFIER: US 3648253 A

TITLE: PROGRAM SCHEDULER FOR PROCESSING SYSTEMS

DATE-ISSUED: March 7, 1972

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mullery; Alvin P.	Chappaqua	NY		
Zurcher, Jr.; Frank W.	Yorktown Heights	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			
Burroughs Corporation	Detroit	MI			

APPL-NO: 04/ 883983 [PALM]
DATE FILED: December 10, 1969

INT-CL: [] G06f 9/18

US-CL-ISSUED: 340/172.5
US-CL-CURRENT: 709/100

FIELD-OF-SEARCH: 340/172.5, 235/157

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL

said scheduler including:

a determination unit for identifying, when the execution of any one of said plurality of hardware engines is terminated, a task allocated to said execution-terminated hardware engine on the basis of said task management information; and

a state controller which performs, upon being activated by said determination unit, a function of updating said state information so that said identified task makes a state transition from said third state to said first state.

5. The microcontroller according to claim 4, wherein said state controller performs, upon being activated by said determination unit, a function of updating said state information so that a task under execution makes a state transition from said second state to said first state.

6. The microcontroller according to claim 4, wherein said scheduler further includes a priority encoder for selecting, on the basis of said task management information, a task having the highest execution priority in all tasks that are in said first state as a task to be run next.

7. The microcontroller according to claim 6, wherein said state controller further performs a function of updating said state information so that said task selected by said priority encoder makes a state transition from said first state to said second state.

8. The microcontroller according to claim 1, wherein said task management table has a region in which to save resources of said processor concerning a task that was run prior to the occurrence of the aforesaid task switching.

9. The microcontroller according to claim 1 further comprising a plurality of register files for use by said plurality of hardware engines as mutually independent working areas.

10. The microcontroller according to claim 1 further comprising a register file used to store a setting parameter common to at least two of said plurality of hardware engines.

11. A data processing system comprising:

a plurality of hardware engines for executing respective data processes; and

a microcontroller for controlling said plurality of hardware engines;

said microcontroller including:

a processor for sequentially executing a plurality of tasks in accordance with programmed instructions, said processor operating in conjunction with said plurality of hardware engines;

a task management table for storing task management information including (i) state information representative of the execution status of each said task, (ii) priority information representative of the execution priority of each said task, and (iii) allocation information representative of the allocation of said plurality of tasks to hardware engines; and

a scheduler for allowing, on the basis of said task management information, said processor to switch between tasks, wherein each said plurality of hardware engines starts execution of a data process upon the activation by said processor and, if said data process is terminated, informs said scheduler of the termination of execution, and said scheduler allows said processor to switch between tasks if the termination of execution of any one of said

hardware engines is detected.

12. The data processing system according to claim 11, wherein each said task can be in one of a first state representative of an execution wait status, a second state representative of a running status, and a third state representative of a wait status awaiting the termination of execution of a hardware engine allocated thereto.

13. The data processing system according to claim 11, wherein each of said plurality of hardware engines is a subprocessing core for MPEG image data encoding.

14. A task switching control method including allocating one or more tasks to corresponding hardware engines and controlling, based on information about such task/hardware engine allocation, task switching by the use of a scheduler, wherein:

each said hardware engines starts execution of a data process upon the activation by a processor and, if said data process is terminated, informs said scheduler of the termination of execution;

each said task can be in one of a first state representative of an execution wait status, a second state representative of a running status, and a third state representative of a wait status awaiting the termination of execution of a hardware engine allocated thereto; and

when the execution of any one of said hardware engines is terminated, said scheduler changes the state of a task allocated to said executon-terminated hardware engine from said third state to said first state so as to allow said processor to switch between tasks.

15. The method according to claim 14, wherein when the execution of a hardware engine is terminated, the state of a task under execution is changed from said second state to said first state.

WEST

End of Result Set

Generate Collection Print

L4: Entry 28 of 28 File: USPT Mar 7, 1972

US-PAT-NO: 3648253
DOCUMENT-IDENTIFIER: US 3648253 A

TITLE: PROGRAM SCHEDULER FOR PROCESSING SYSTEMS

DATE-ISSUED: March 7, 1972

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mullery; Alvin P.	Chappaqua	NY		
Zurcher, Jr.; Frank W.	Yorktown Heights	NY		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			
Burroughs Corporation	Detroit	MI			

APPL-NO: 04/ 883983 [PALM]
DATE FILED: December 10, 1969

INT-CL: [] G06f 9/18

US-CL-ISSUED: 340/172.5
US-CL-CURRENT: 709/100

FIELD-OF-SEARCH: 340/172.5, 235/157

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3286239	November 1966	Thompson et al.	340/172.5
☐	3312951	April 1967	Hertz	340/172.5
☐	3333252	July 1967	Shimabukuro	340/172.5
☐	3359544	December 1967	Macon et al.	340/172.5
☐	3363234	January 1968	Erickson et al.	340/172.5
☐	3399384	August 1968	Crockett et al.	340/172.5
☐	3421150	January 1969	Quosig et al.	340/172.5
☐	3449722	June 1969	Tucker	340/172.5
☐	3491339	January 1970	Schramel	340/172.5
☐	3496551	February 1970	Driscoll et al.	340/172.5
☐	3530438	September 1970	Mellen et al.	340/172.5

ART-UNIT: 237

PRIMARY-EXAMINER: Shaw; Gareth D.

ASSISTANT-EXAMINER: Chirlin; Sydney R.

ABSTRACT:

A program scheduler is provided for use with a multiprocessor system or its equivalent, such as a multiprogrammed processor unit, and the program scheduler receives tasks to be executed, schedules them for assignment, allots a task to each processor and interrupts the processors to assign new tasks. The program scheduler includes a plurality of buckets or tables where task words are stored, and associated with each task word is a T.sub.e field which specifies the estimated processor time required to complete the task and a T.sub.d field which indicates the time remaining before the task must be completed. The ratio T.sub.e /T.sub. d provides an indication of the need of each task word for processor service since the need for such service becomes more urgent as the ratio approaches 1. A scheduling algorithm periodically recalculates the service ratio and shifts tasks, if need be, from one table to another whereby tasks with a similar service ratio are stored in a common table. Task words within a given table are divided into classes according to the length of time a task has not received service. An allocation algorithm allots tasks to processors from the older classes first and proceeds in sequence through the various classes to the latest classes. Both the scheduling algorithm and the allocation algorithm service all tables in the program scheduler, but the tables with higher service ratios are serviced more often by each algorithm than tables with lower service ratios. When many task words are awaiting processor service, a given task word receives processor service at a rather low frequency when it has a small service ratio, but it receives processor service at a relatively high frequency as its service ratio approaches 1.

40 Claims, 65 Drawing figures